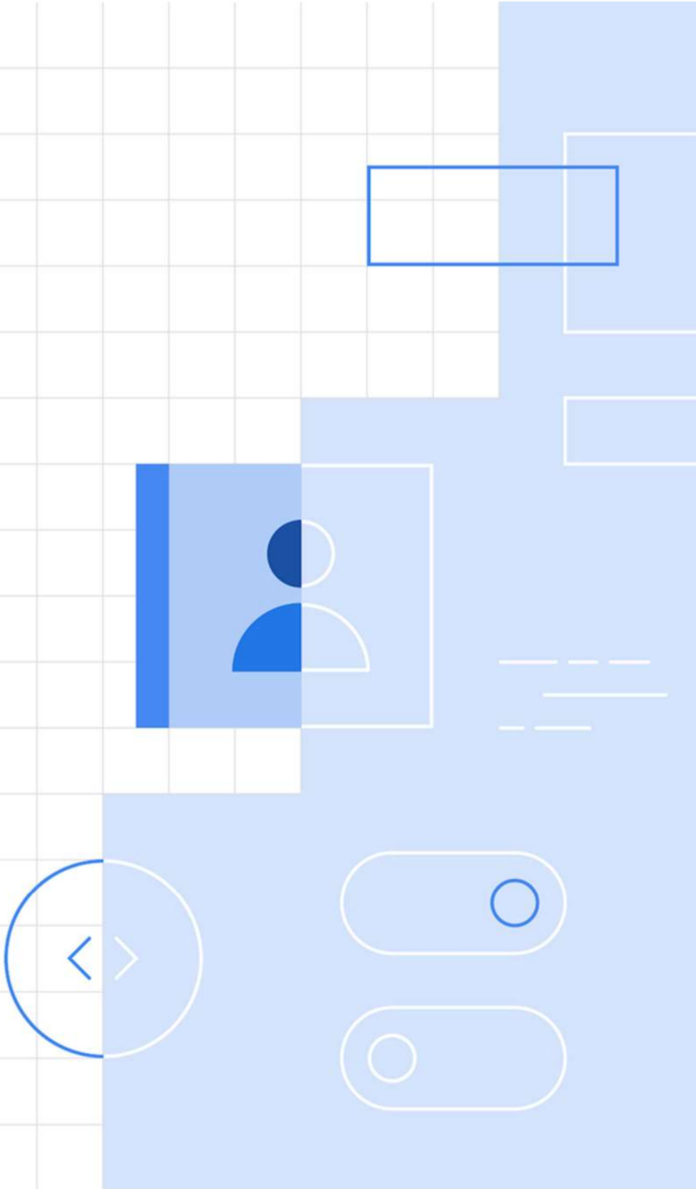
WELCOME !

Developer Student Clubs

Front-end Development: CSS

Benjamin Ogbonna

DSC Uniabuja



Google Developers

OUTLINE

- Introduction to CSS
- Applying CSS to HTML
- Selectors, Properties and Values
- CSS Colors and Backgrounds
- CSS Margins, Padding and Border
- CSS Text and Font Properties
- CSS Positioning

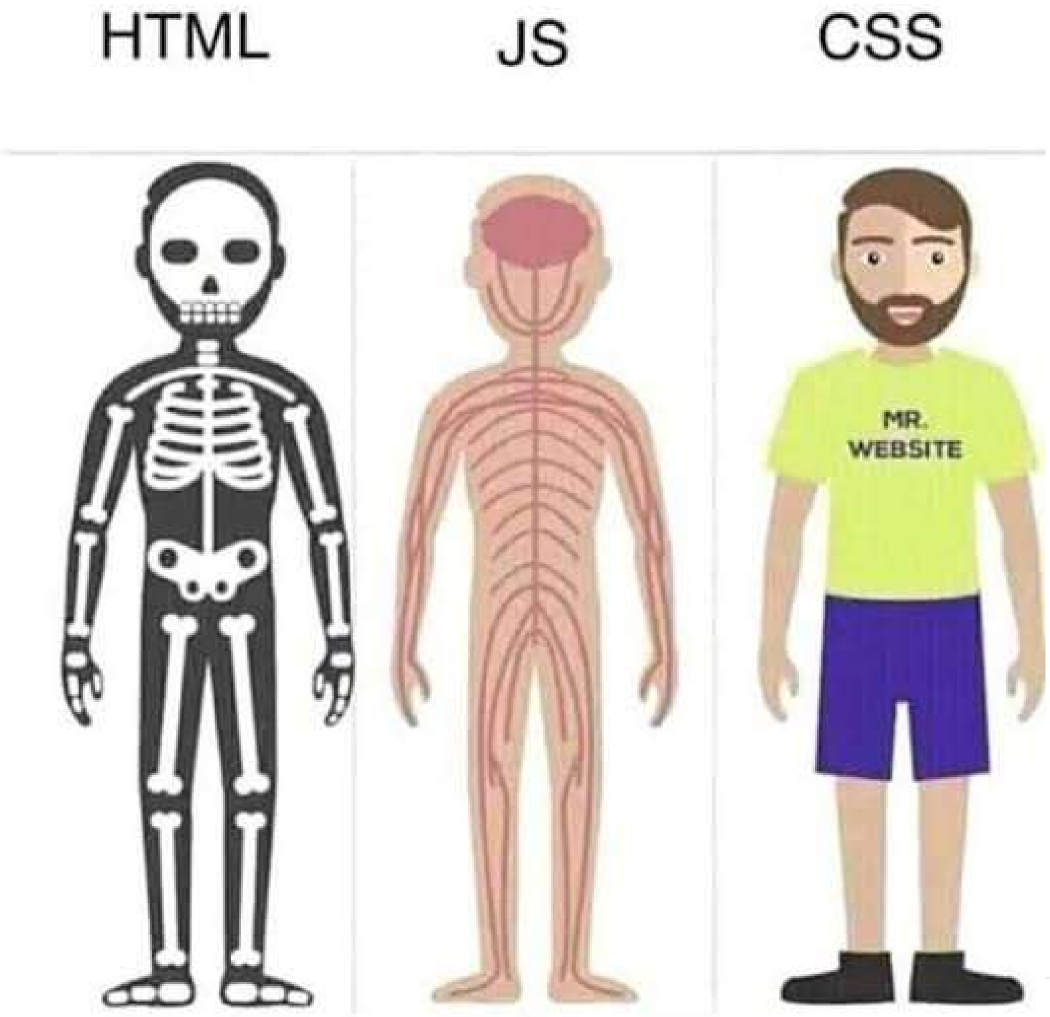
INTRODUCTION

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. The separation can improve content accessibility, provide more flexibility and control in the specification of presentation. CSS is a stylesheet language that describes the presentation of an HTML (or XML) document. CSS describes how elements must be rendered on screen, on paper, or in other media. CSS has a simple Syntax and uses a number of English keywords to specify the names of various style properties. A style sheet consist of a list of rules. Each rule or rule-set consists of one or more selectors and declaration block.

What is CSS?

- **CSS** stands for **Cascading Style Sheets**
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- CSS saves a lot of work. It can control the layout of multiple web pages all at once.
- External stylesheets are stored in **CSS files**

ANALOGY



Applying CSS to HTML

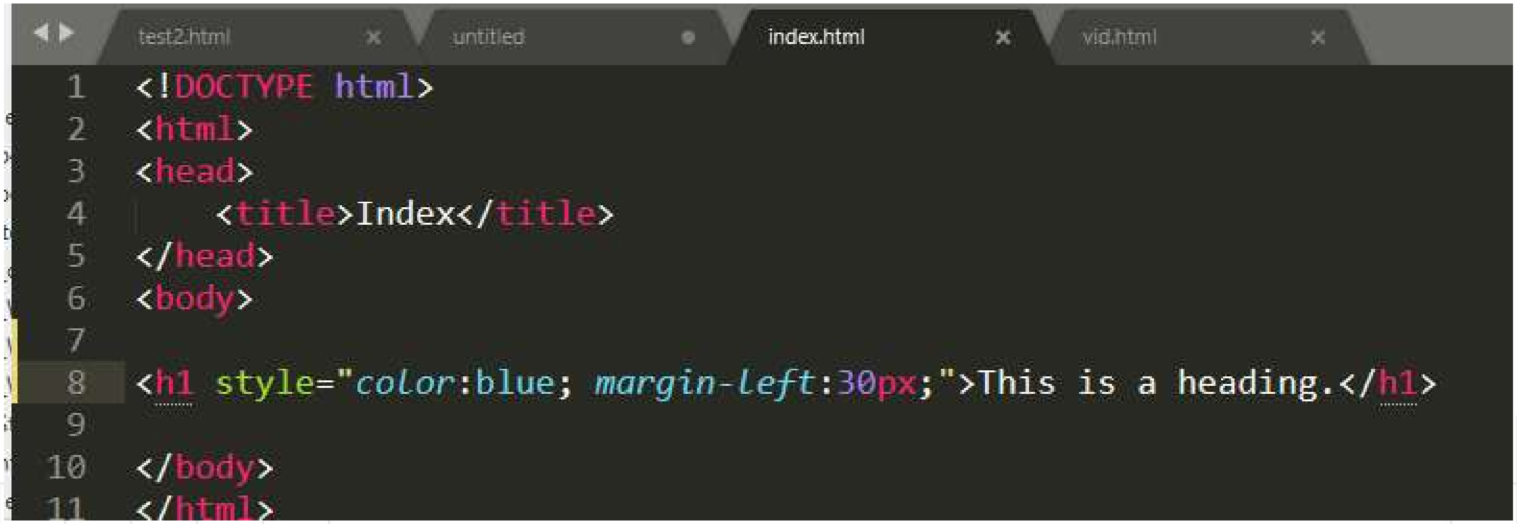
When a browser reads a style sheet, it will format the HTML document according to the information in the style sheet.

There are three ways of inserting a style sheet:

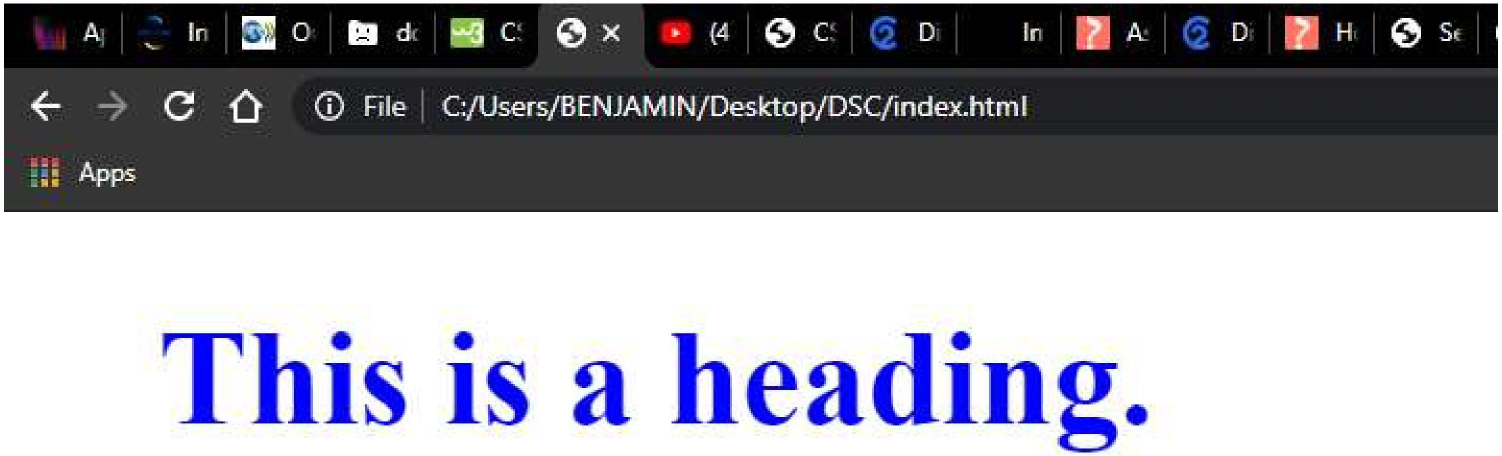
- **Inline Style**
- **Internal (Embedded) style sheet**
- **External style sheet**

INLINE STYLE:

An inline style may be used to apply a unique style for a single element. To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property. The example below shows how to change the color and the left margin of a <h1> element:



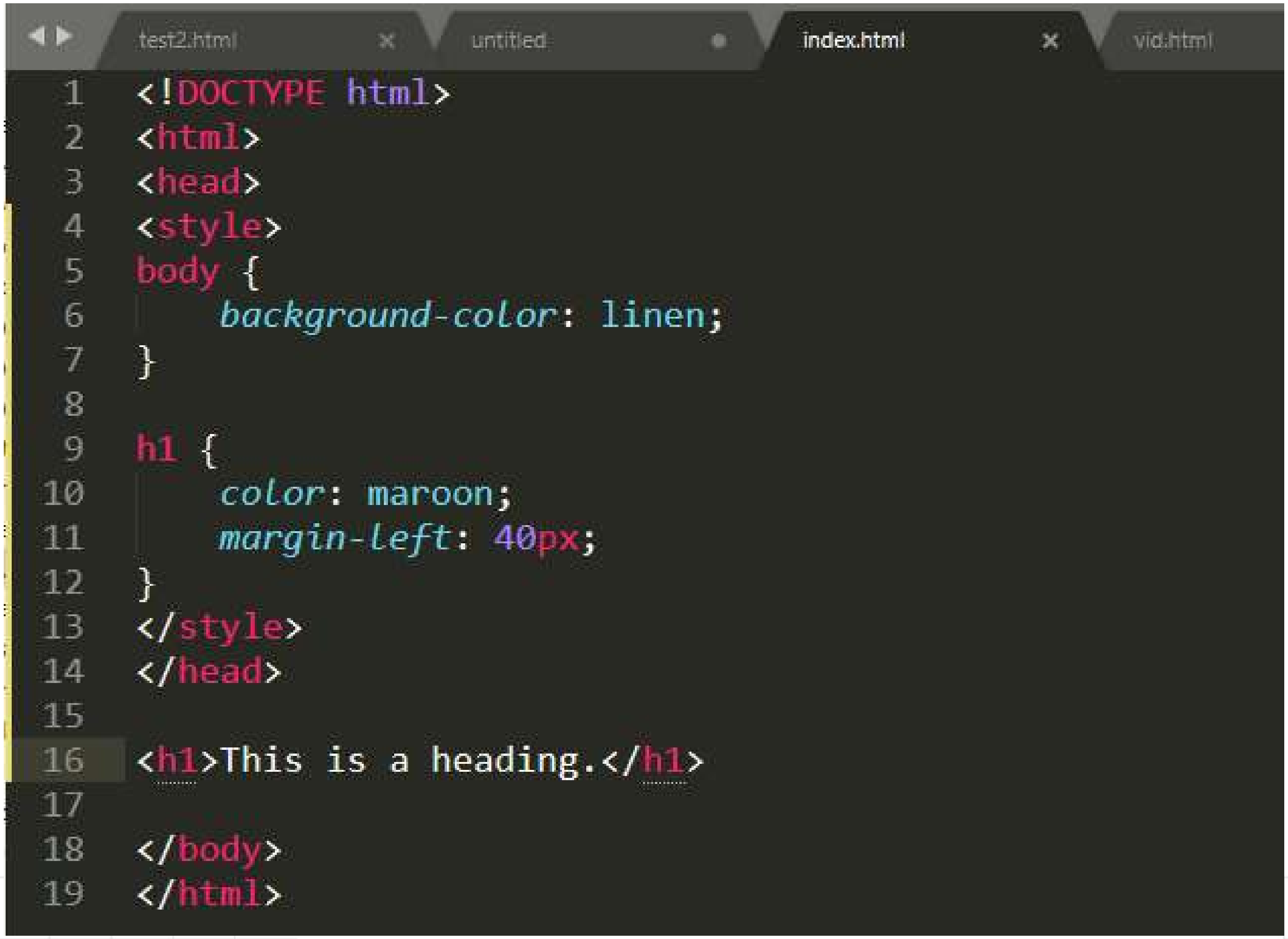
```html
<!DOCTYPE html>
<html>
<head>
    <title>Index</title>
</head>
<body>

<h1 style="color:blue; margin-left:30px;">This is a heading.</h1>

</body>
</html>
```



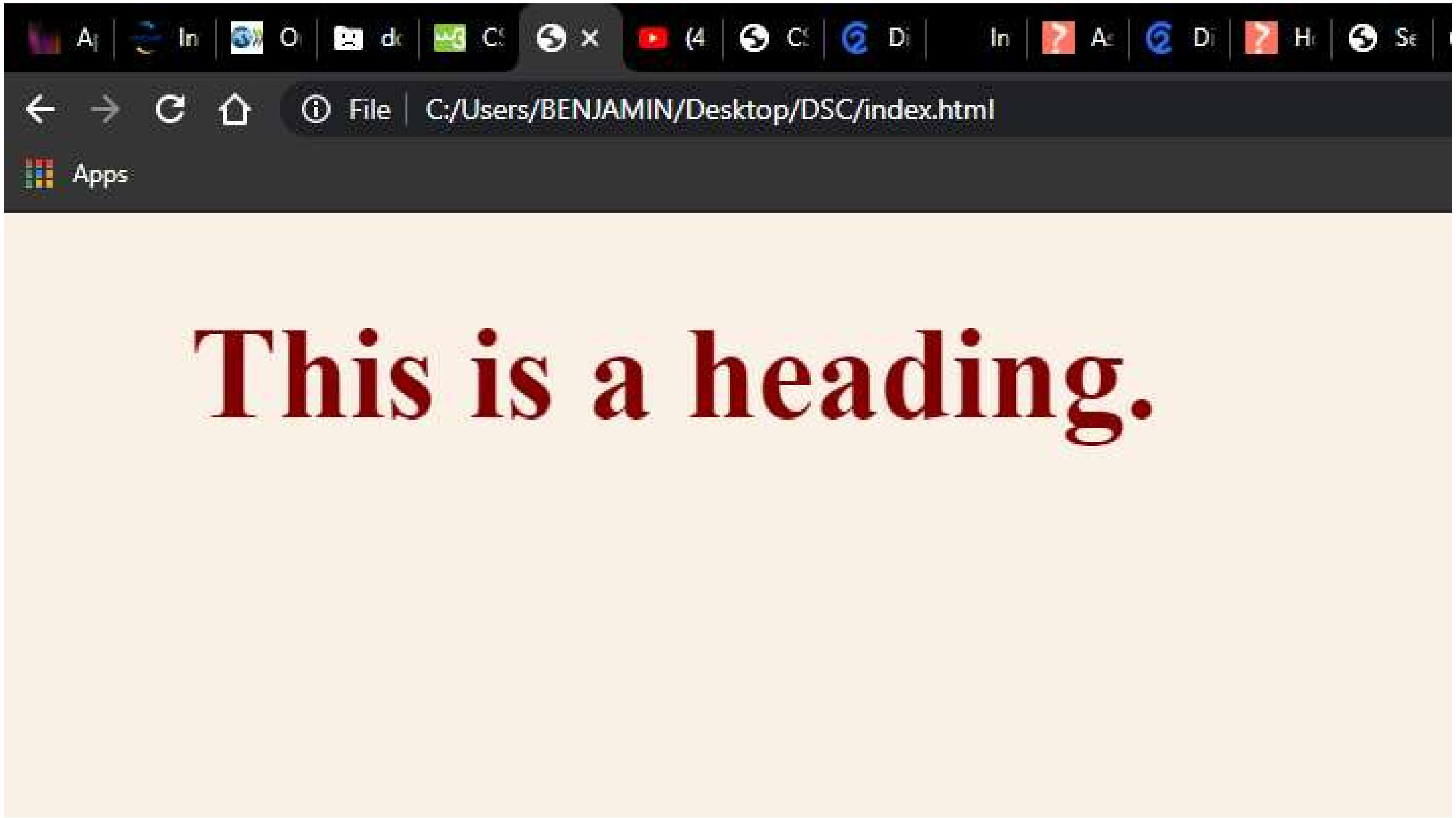
This is a heading.

INTERNAL STYLE SHEET:

Internal Style Sheet
An internal style sheet may be used if one single page has a unique style. Internal styles are defined within the <style> element, inside the <head> section of an HTML page:



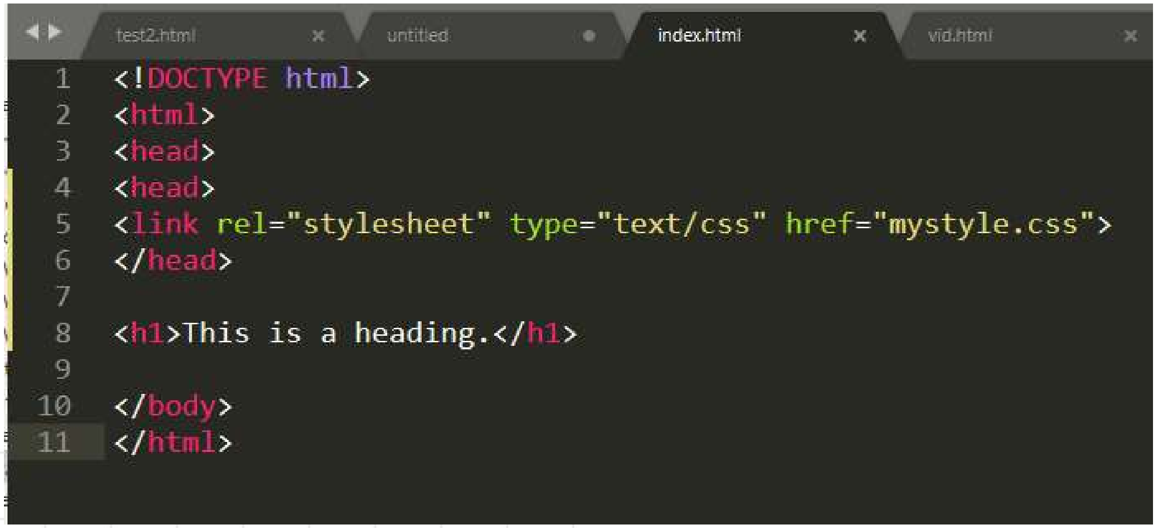
```html
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-color: linen;
}

h1 {
    color: maroon;
    margin-left: 40px;
}
</style>
</head>

<h1>This is a heading.</h1>

</body>
</html>
```



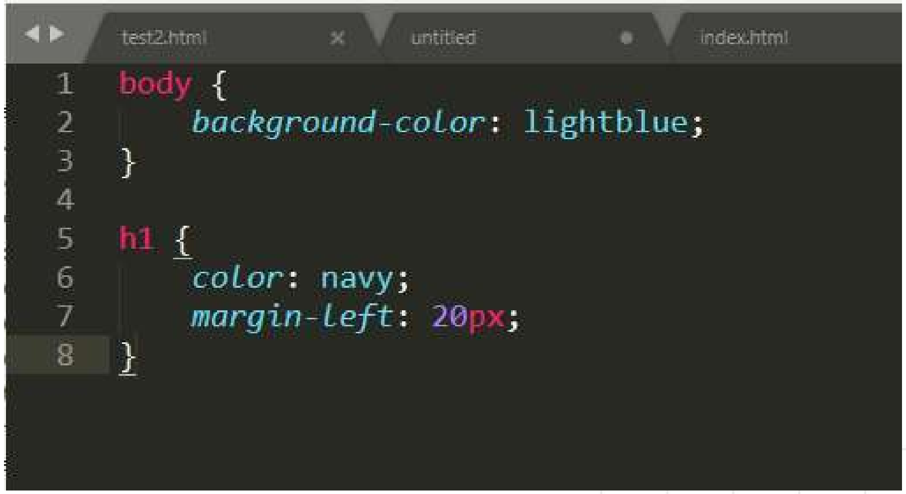
This is a heading.

EXTERNAL STYLE SHEET:

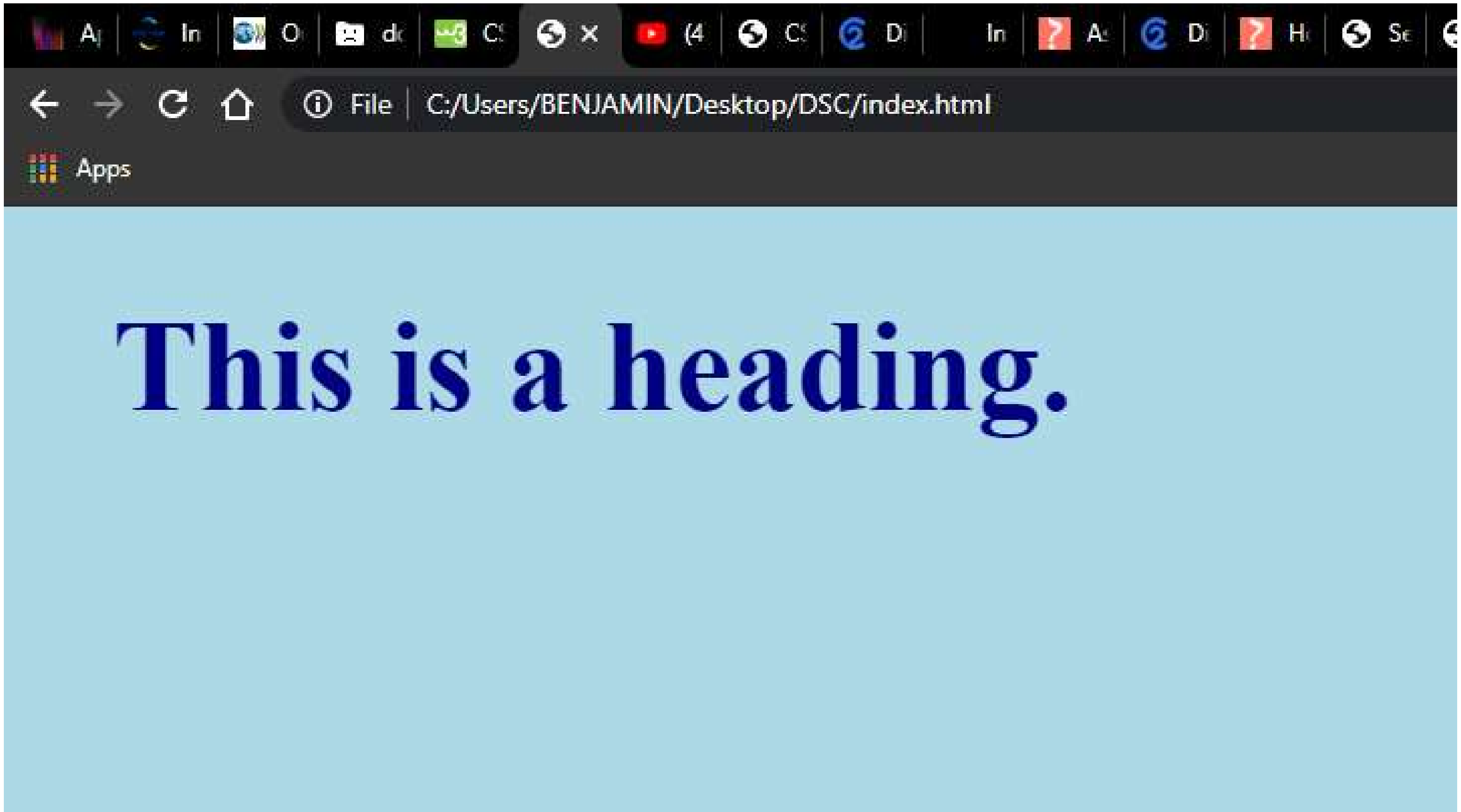
With an external style sheet, you can change the look of an entire website by changing just one file!
Each page must include a reference to the external style sheet file inside the <link> element. The <link> element goes inside the <head> section:



```html
<!DOCTYPE html>
<html>
<head>
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>

<h1>This is a heading.</h1>

</body>
</html>
```



```css
body {
    background-color: lightblue;
}

h1 {
    color: navy;
    margin-left: 20px;
}
```



This is a heading.

CASCADING ORDER:

What style will be used when there is more than one style specified for an HTML element?

Generally speaking we can say that all the styles will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest priority:

1. **Inline style (inside an HTML element)**
2. **Internal (Embedded) style sheets**
3. **External**
4. **Browser default**

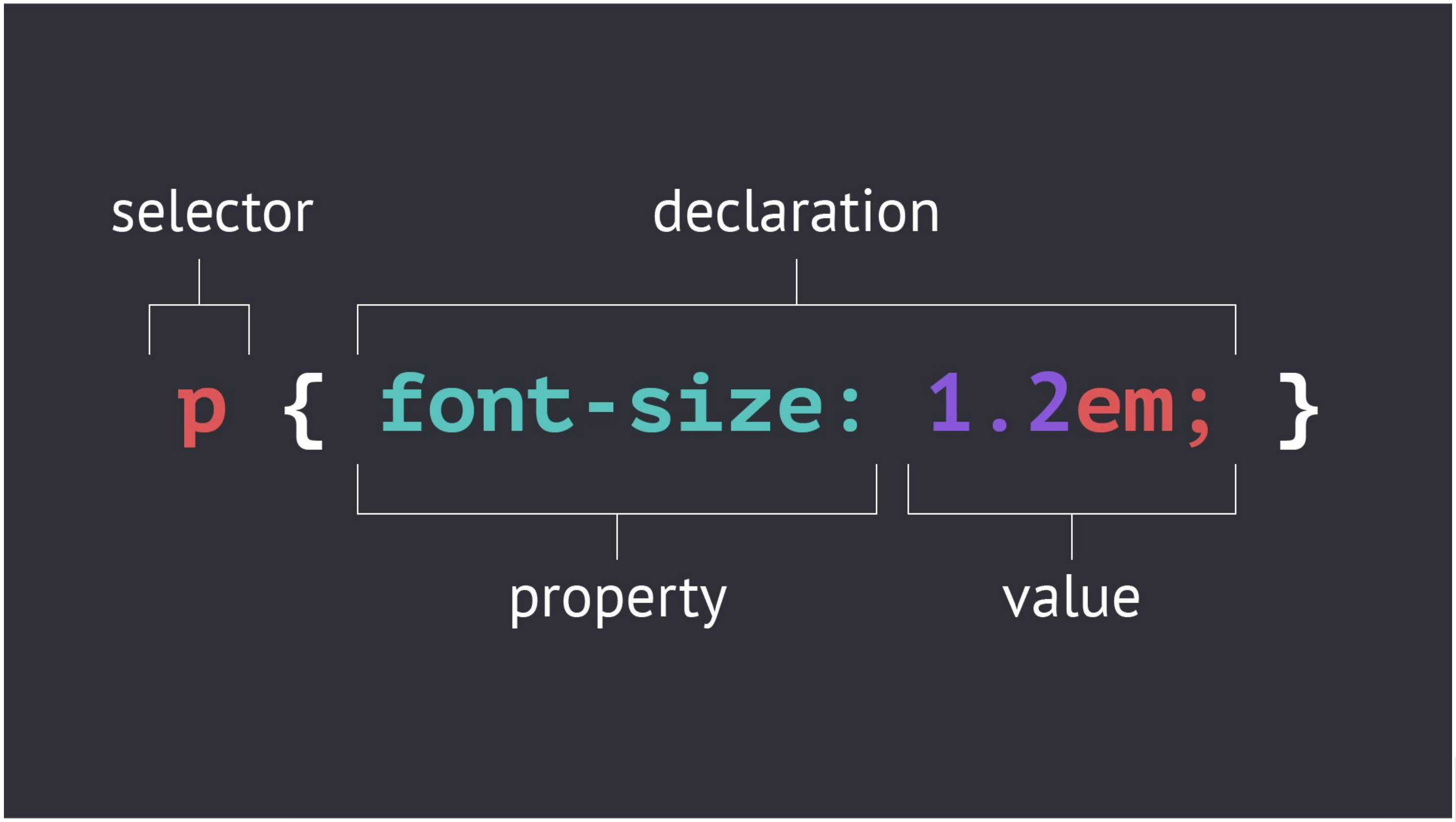
So, an inline style (inside a specific HTML element) has the highest priority, which means that it will override a style defined inside the <head> tag, or in an external style sheet, or a browser default value.

SELECTORS, PROPERTIES AND VALUE:

CSS properties are the styles used on specified **selectors**. They are written before **values** in the **CSS** ruleset and are separated from **property values** by a colon. Different HTML **selectors** and elements have different **properties**. Some **properties** are universal and can be used on every **selector**.

Here are four common properties to work with:

- List properties
- Font properties
- Border properties
- Text properties



selector

declaration

```css
p { font-size: 1.2em; }
```

property

value

CSS COLORS:

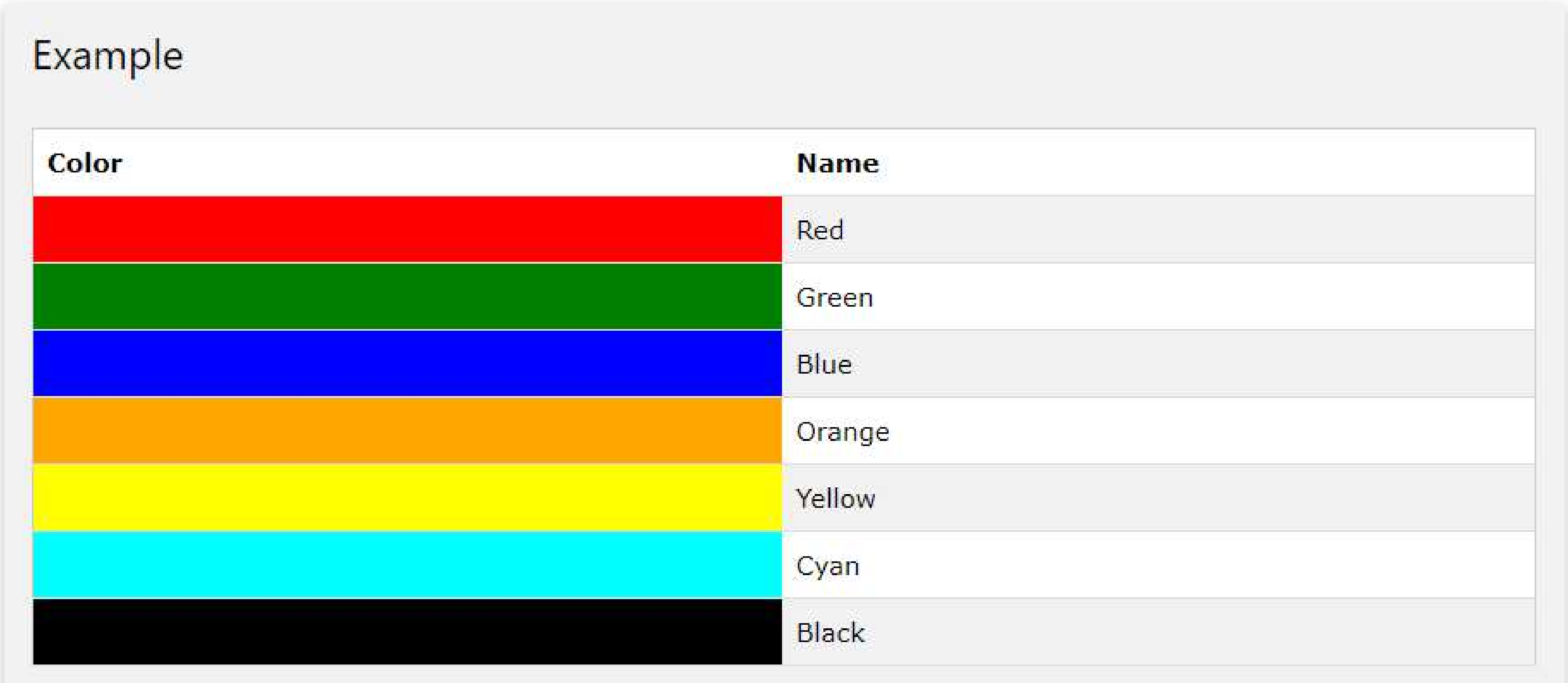
Colors are displayed combining RED, GREEN, and BLUE light.

Colors in CSS are most often specified by:

- a valid color name - like "red"
- an RGB value - like "rgb(255, 0, 0)"
- a HEX value - like "#ff0000“

Note: Color names are case-insensitive: "Red" is the same as "red" or "RED".
HTML and CSS supports 140 standard color names.

Colors set by using color names:



Example

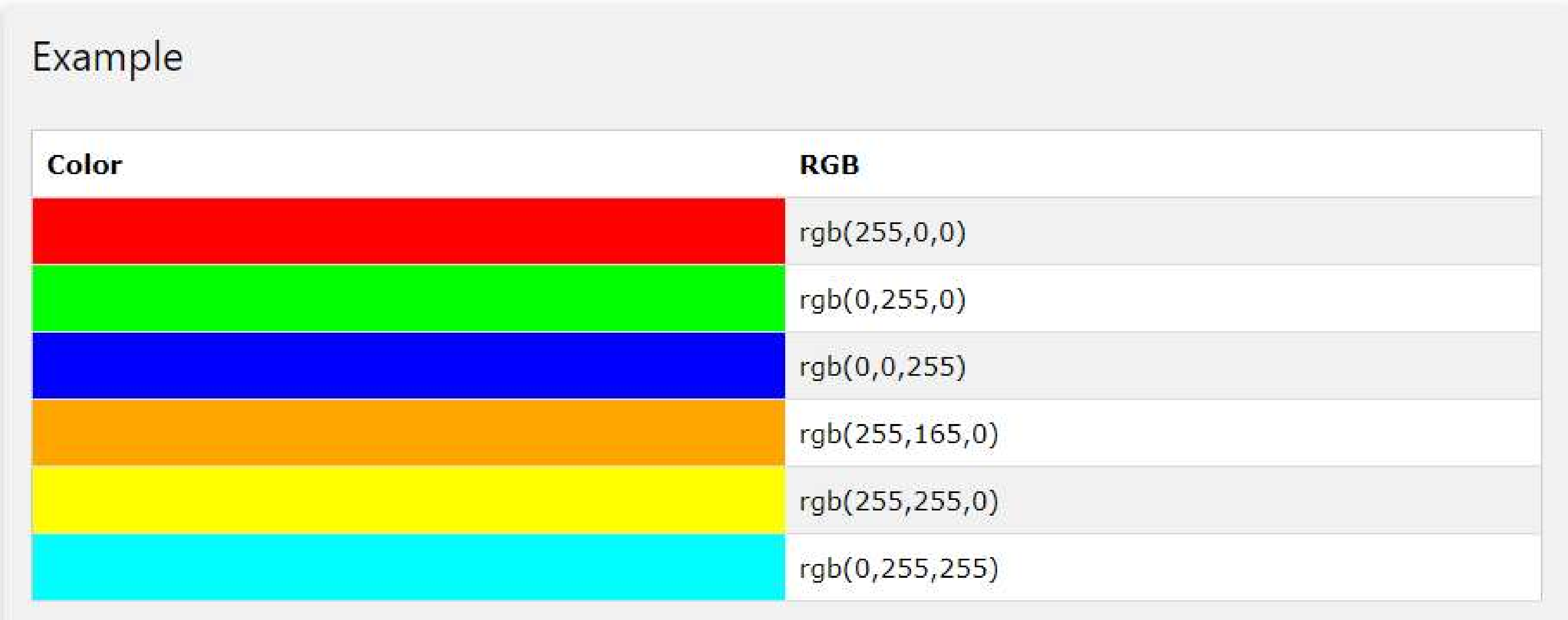
Color	Name
🟥	Red
🟩	Green
🟦	Blue
🟧	Orange
🟨	Yellow
🟦	Cyan
⬛	Black

CSS COLORS:

RGB (Red, Green, Blue)
RGB color values can be specified using this formula: rgb(red, green, blue).

Each parameter (red, green, blue) defines the intensity of the color between 0 and 255.

For example, rgb(255,0,0) is displayed as red, because red is set to its highest value (255) and the others are set to 0. Experiment by mixing the RGB values below:



Example

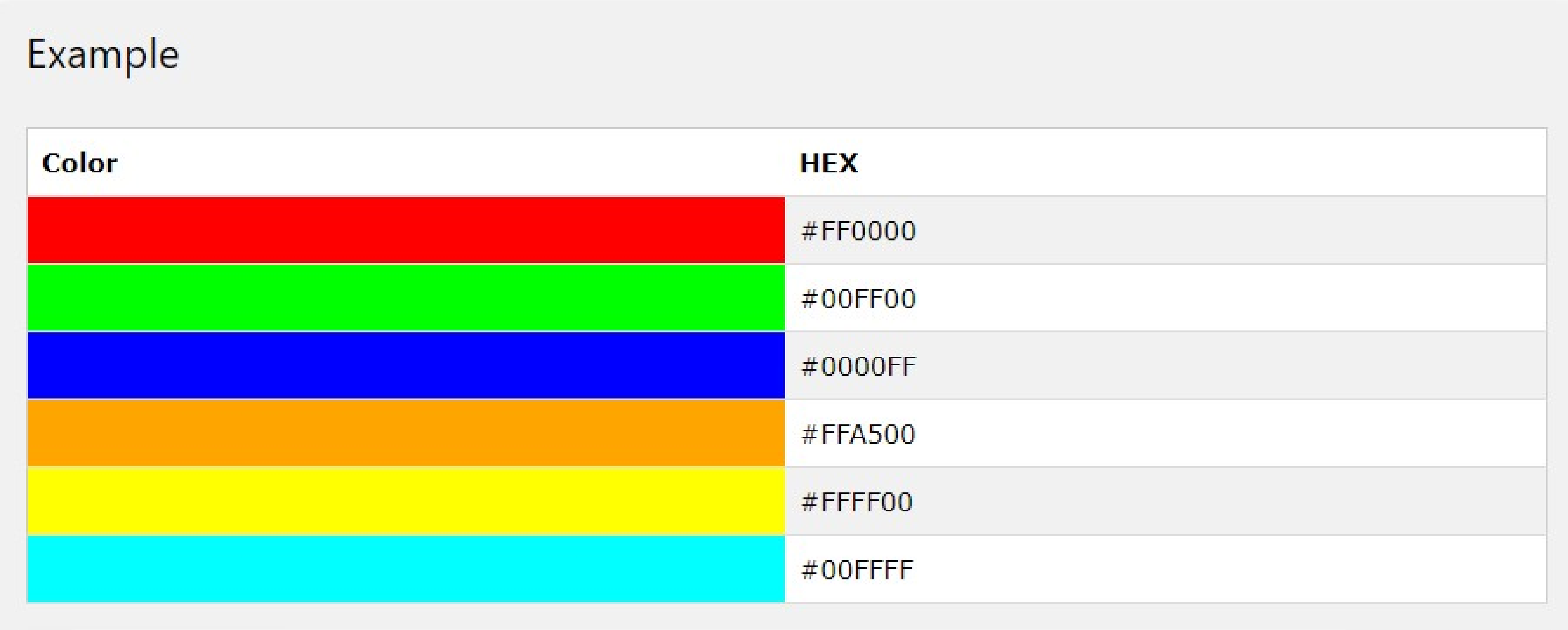
Color	RGB
🟥	rgb(255,0,0)
🟩	rgb(0,255,0)
🟦	rgb(0,0,255)
🟧	rgb(255,165,0)
🟨	rgb(255,255,0)
🟦	rgb(0,255,255)

CSS COLORS:

Hexadecimal Colors

RGB values can also be specified using hexadecimal color values in the form: #RRGGBB, where RR (red), GG (green) and BB (blue) are hexadecimal values between 00 and FF (same as decimal 0-255).

For example, #FF0000 is displayed as red, because red is set to its highest value (FF) and the others are set to the lowest value (00). Note: HEX values are case-insensitive: "#ff0000" is the same as "FF0000".



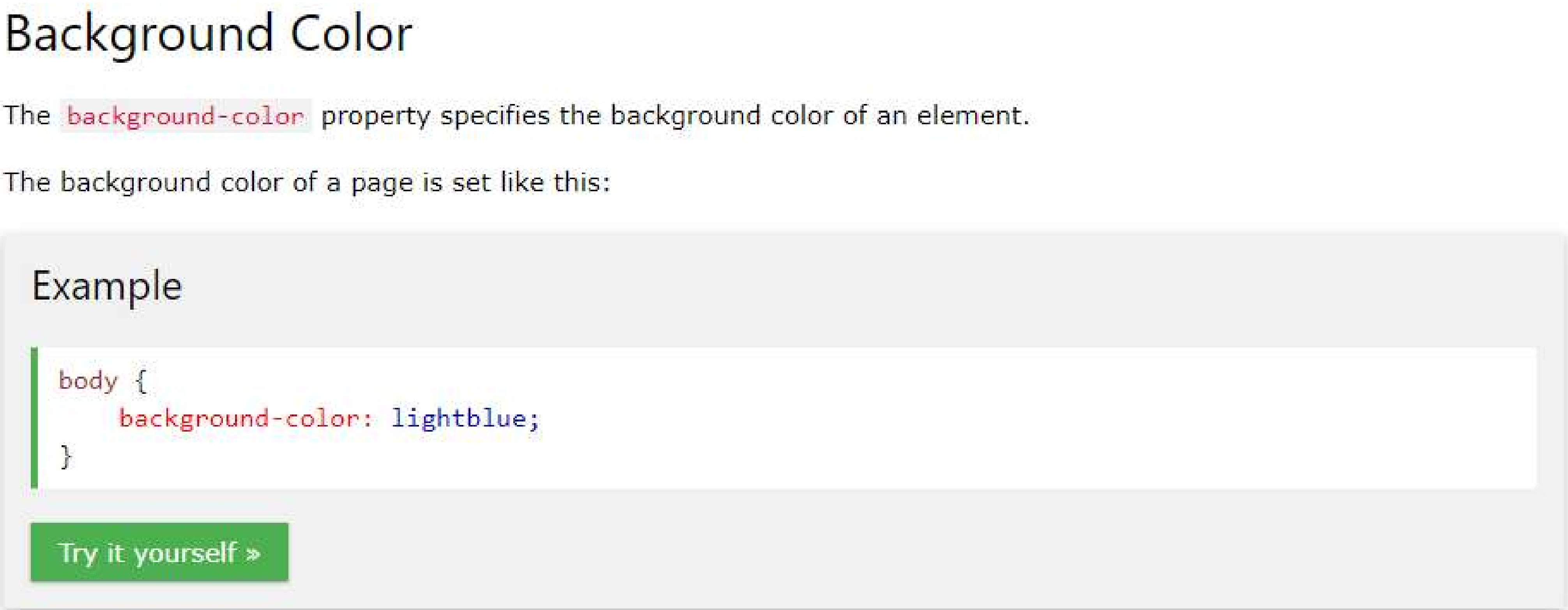
Example

Color	HEX
🟥	#FF0000
🟩	#00FF00
🟦	#0000FF
🟧	#FFA500
🟨	#FFFF00
🟦	#00FFFF

CSS BACKGROUNDS:

The CSS background properties are used to define the background effects for elements.
CSS background properties:

- background-color
- background-image
- background-repeat
- background-position



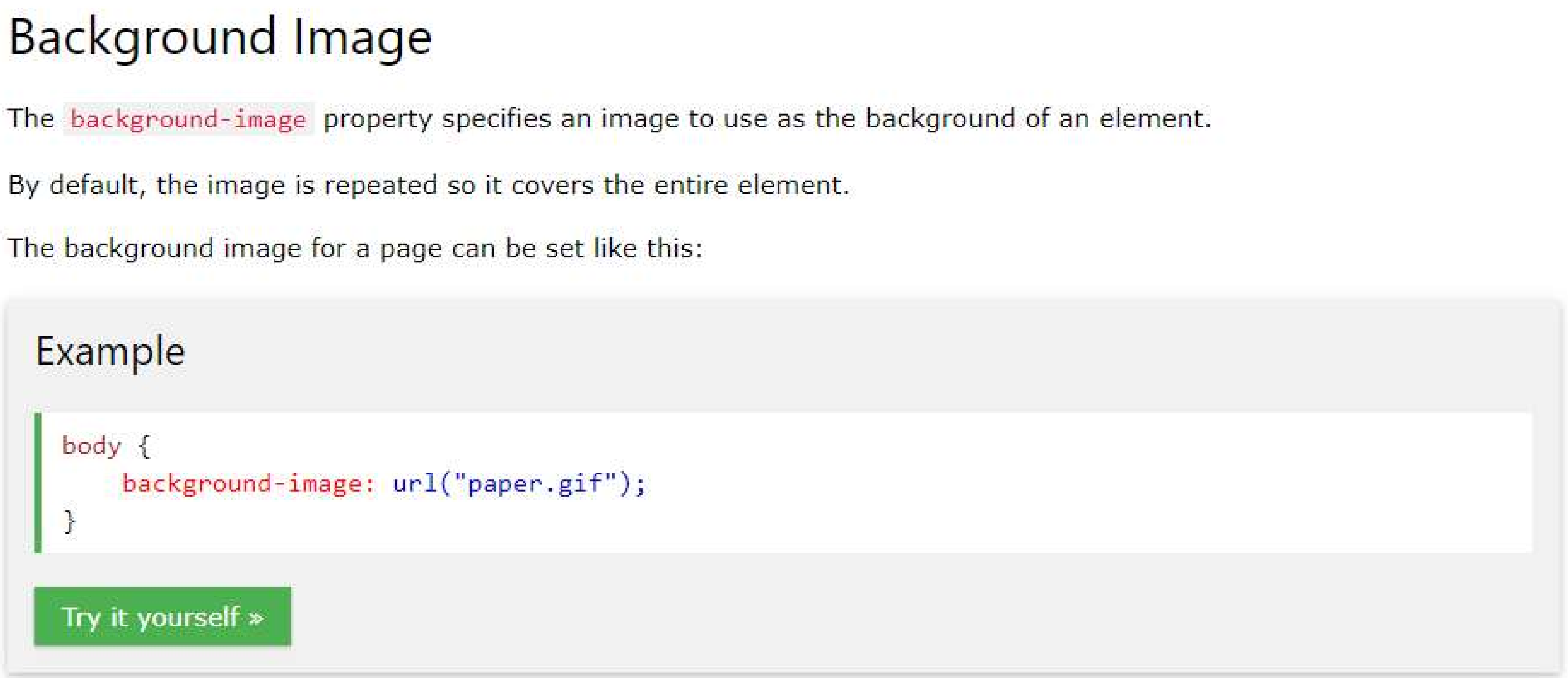
Background Color

The `background-color` property specifies the background color of an element.

The background color of a page is set like this:

Example

```css
body {
    background-color: lightblue;
}
```



Background Image

The `background-image` property specifies an image to use as the background of an element.

By default, the image is repeated so it covers the entire element.

The background image for a page can be set like this:

Example

```
body {
    background-image: url("paper.gif");
}
```

CSS BORDERS:

CSS Border Properties

The CSS border properties allow you to specify the style, width, and color of an element's border. This element has a groove border that is 10px wide and green.

CSS BORDERS:

Border Style

The border-style property specifies what kind of border to display.

The following values are allowed:

dotted - Defines a dotted border

dashed - Defines a dashed border

solid - Defines a solid border

double - Defines a double border

groove - Defines a 3D grooved border. The effect depends on the border-color value

ridge - Defines a 3D ridged border. The effect depends on the border-color value

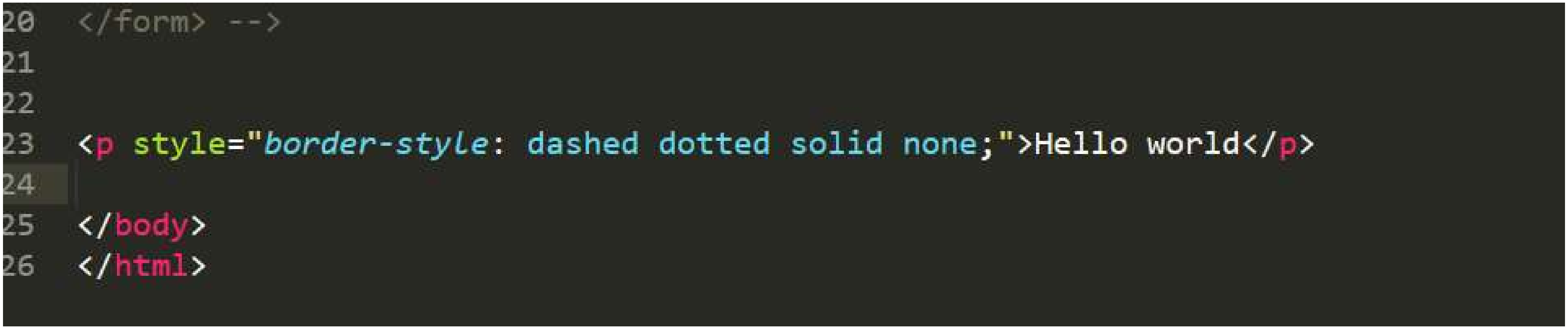
inset - Defines a 3D inset border. The effect depends on the border-color value

outset - Defines a 3D outset border. The effect depends on the border-color value

none - Defines no border

hidden - Defines a hidden border

Note: The border-style property can have from one to four values (for the top border, right border, bottom border, and the left border).



```html
</form> -->


<p style="border-style: dashed dotted solid none;">Hello world</p>

</body>
</html>
```

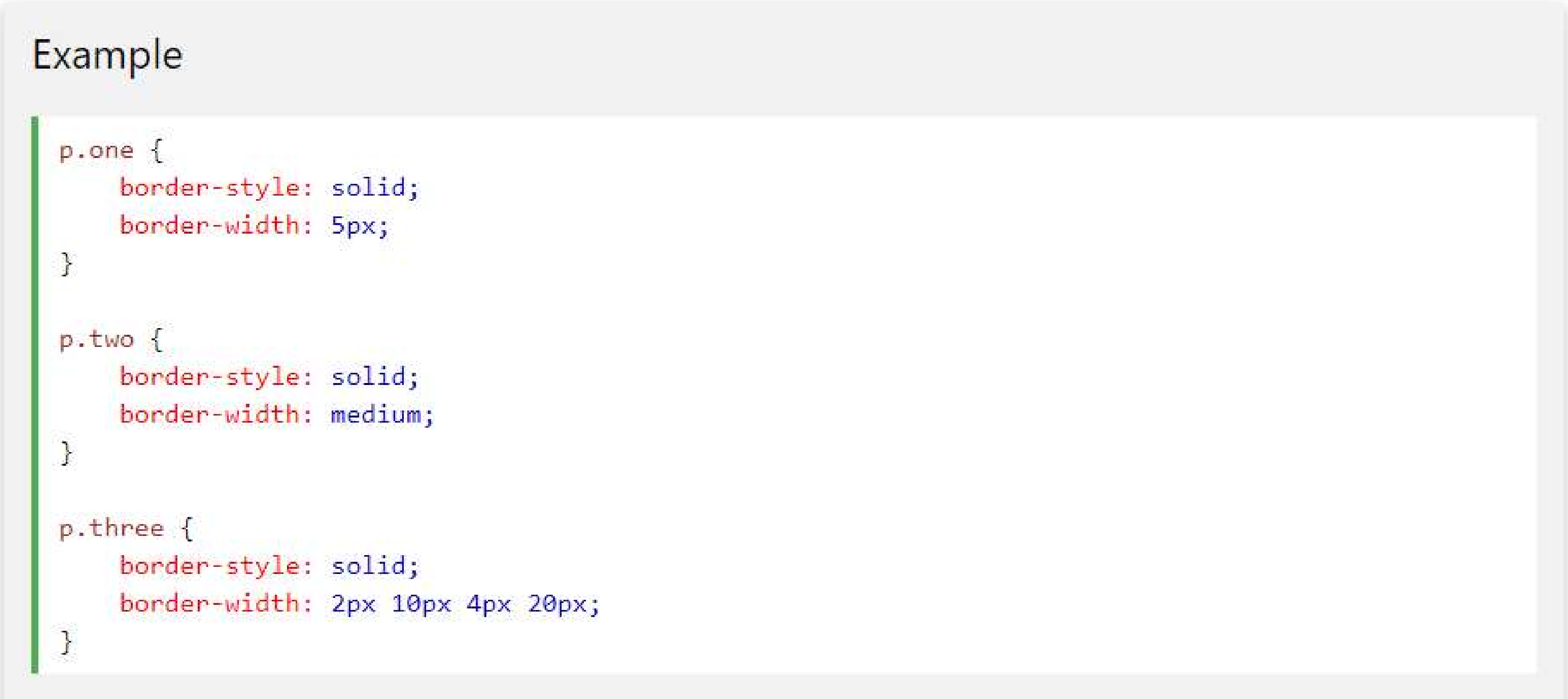
CSS BORDERS:

Border width

The border-width property specifies the width of the four borders.

The width can be set as a specific size (in px, pt, cm, em, etc) or by using one of the three pre-defined values: thin, medium, or thick.

The border-width property can have from one to four values (for the top border, right border, bottom border, and the left border).



Example

```
p.one {
    border-style: solid;
    border-width: 5px;
}

p.two {
    border-style: solid;
    border-width: medium;
}

p.three {
    border-style: solid;
    border-width: 2px 10px 4px 20px;
}
```

CSS BORDERS:

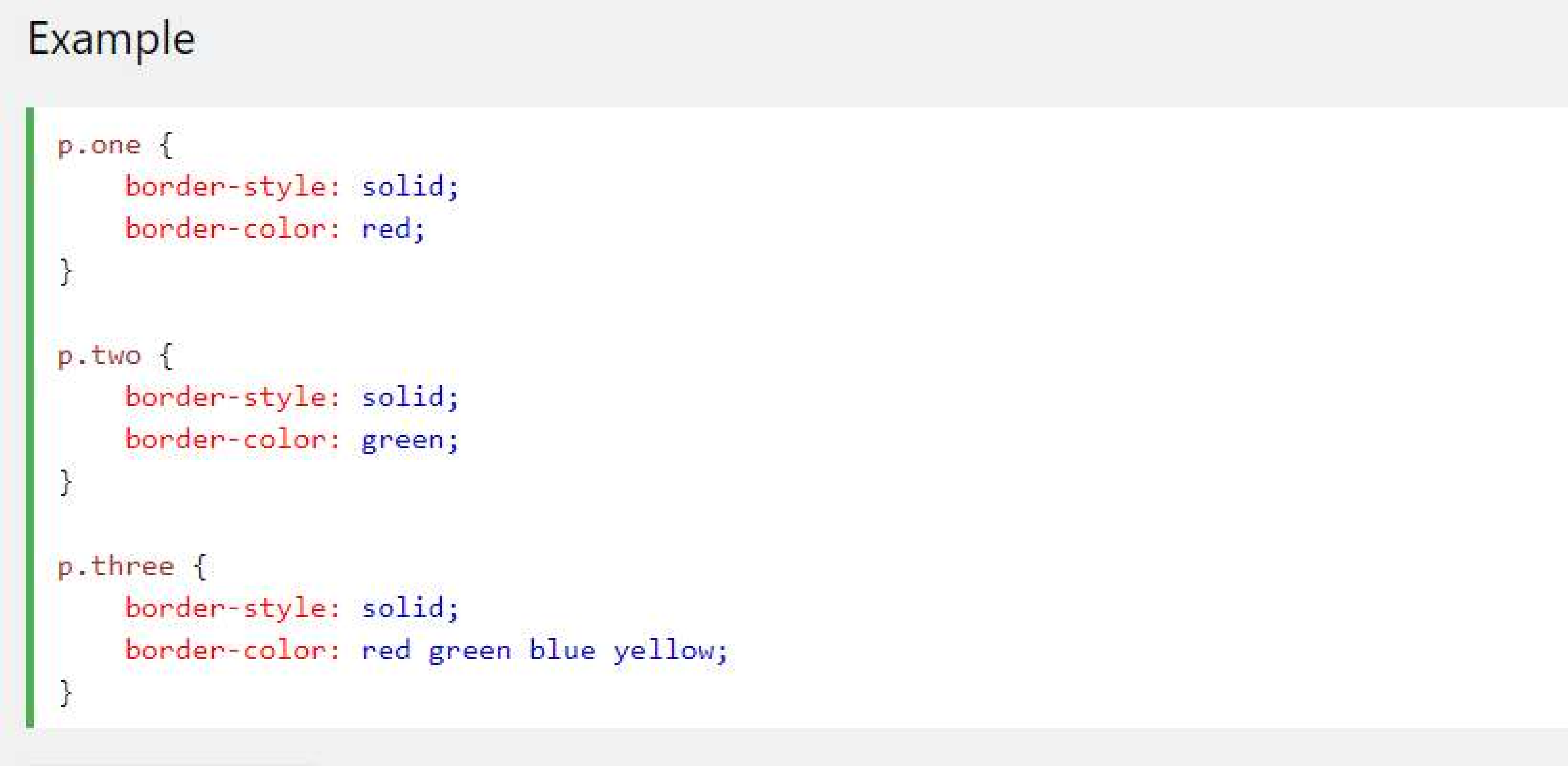
Border Color

The border-color property is used to set the color of the four borders.

The color can be set by:

- name - specify a color name, like "red"
- Hex - specify a hex value, like "#ff0000"
- RGB - specify a RGB value, like "rgb(255,0,0)"
- Transparent

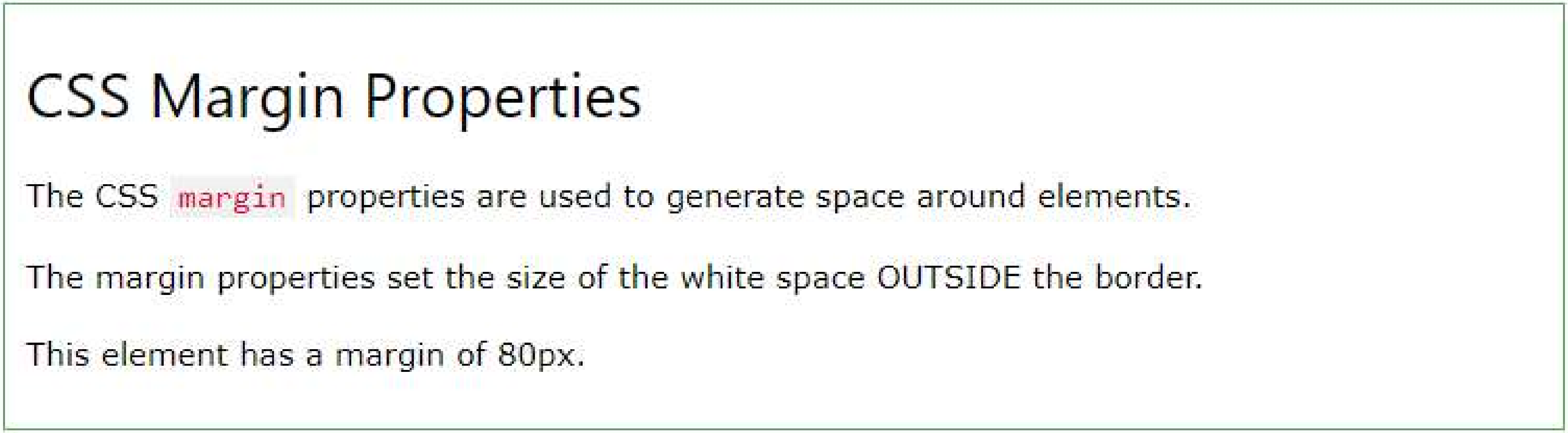
The border-color property can have from one to four values (for the top border, right border, bottom border, and the left border). If border-color is not set, it inherits the color of the element.



Example

```
p.one {
    border-style: solid;
    border-color: red;
}

p.two {
    border-style: solid;
    border-color: green;
}

p.three {
    border-style: solid;
    border-color: red green blue yellow;
}
```

CSS MARGIN:



CSS Margin Properties

The CSS `margin` properties are used to generate space around elements.

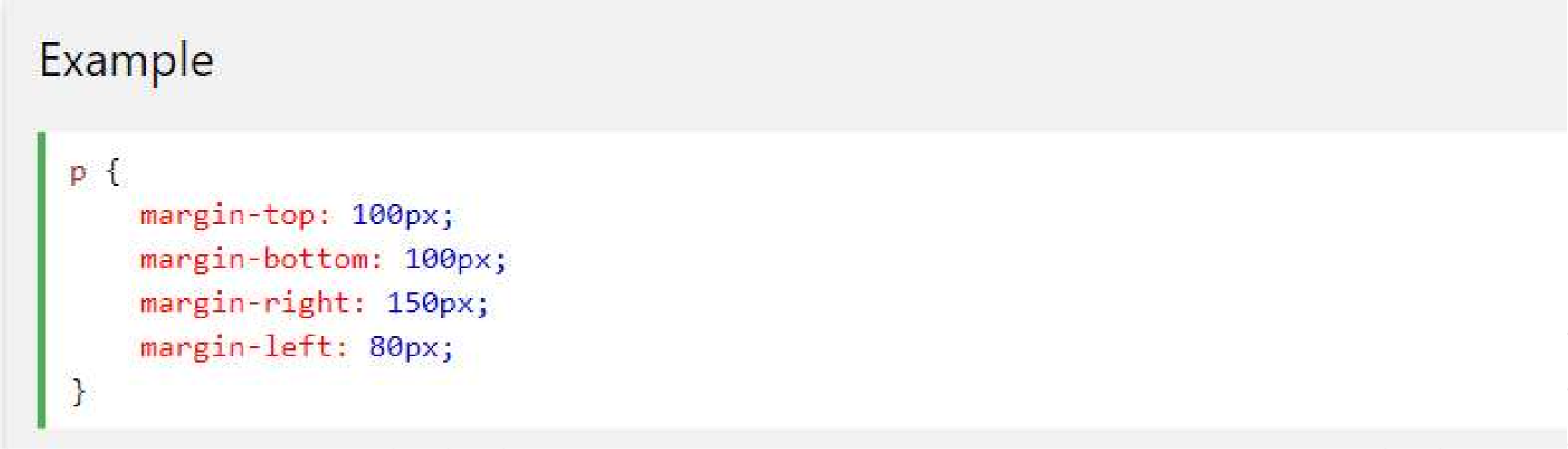
The margin properties set the size of the white space OUTSIDE the border.

This element has a margin of 80px.

The CSS margin properties set the size of the white space OUTSIDE the border.
Note Note: The margins are completely transparent - and cannot have a background color!
With CSS, you have full control over the margins. There are CSS properties for setting the margin for each side of an element (top, right, bottom, and left).

All the margin properties can have the following values:

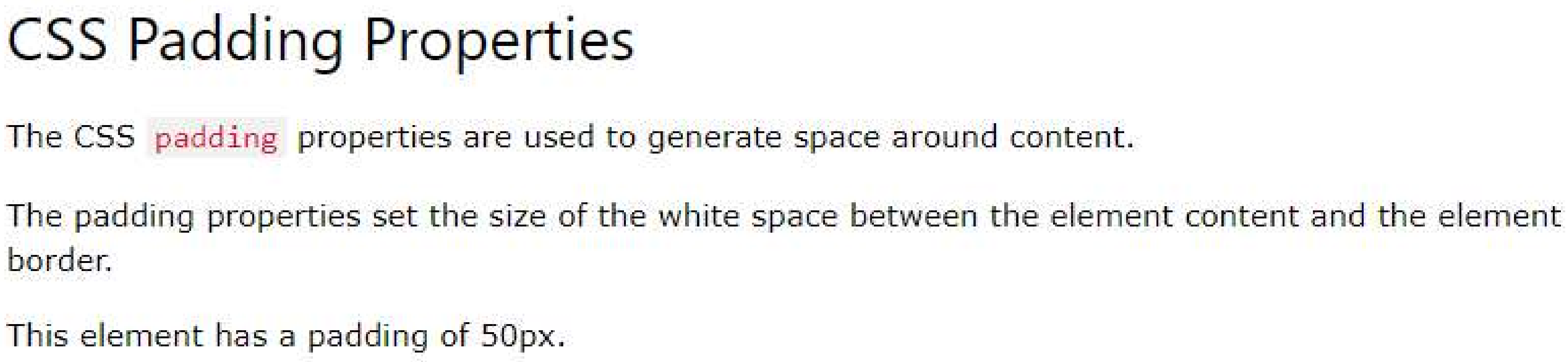
auto - the browser calculates the margin
length - specifies a margin in px, pt, cm, etc.
% - specifies a margin in % of the width of the containing element
inherit - specifies that the margin should be inherited from the parent element



Example

```
p {
    margin-top: 100px;
    margin-bottom: 100px;
    margin-right: 150px;
    margin-left: 80px;
}
```

CSS PADDING:



CSS Padding Properties

The CSS `padding` properties are used to generate space around content.

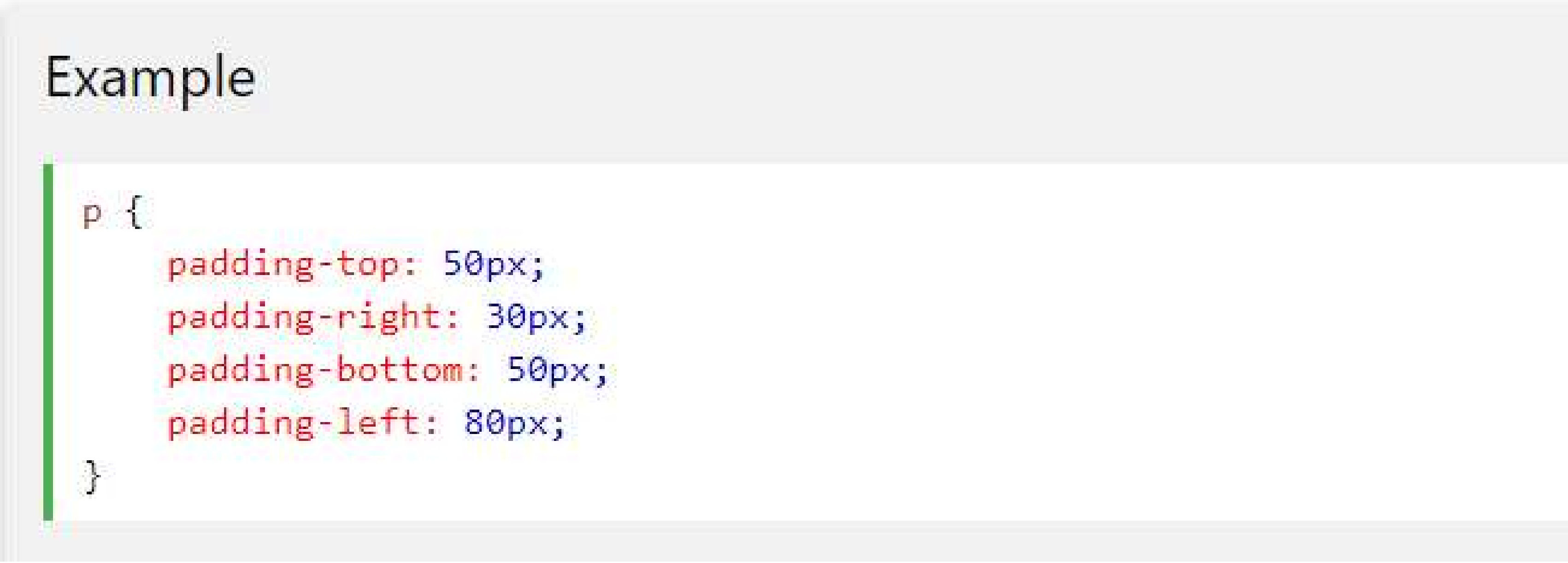
The padding properties set the size of the white space between the element content and the element border.

This element has a padding of 50px.

All the padding properties can have the following values:

- **length -** specifies a padding in px, pt, cm, etc.
- **% -** specifies a padding in % of the width of the containing element
- **inherit -** specifies that the padding should be inherited from the parent element

The following example sets different padding for all four sides of a <p> element:



Example

```
p {
    padding-top: 50px;
    padding-right: 30px;
    padding-bottom: 50px;
    padding-left: 80px;
}
```